Toward a Learning Robot

Tom M. Mitchell
Matthew T. Mason
Alan D. Christiansen

24 January 1989

CMU-CS-89-106

Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

The problem of robot manipulation--of planning how to successfully grasp, push, and pull arbitrary objects to reconfigure them as desired--is an unsolved problem in robotics. We are presently developing a hand-eye system which is intended to begin with simple knowledge about how to manipulate its world, and to itself acquire increasingly refined knowledge about manipulation in specialized contexts. For example, the robot may begin with knowledge such as "to move an object somewhere to the right, push from some point on the left", and may refine the knowledge through experience into more specific, more precise assertions such as "to move an object *which is in contact with a wall* to the right *in a straight line*, push from a point on the left *which is 2/3 of the distance from the wall.*" The learning method is a type of explanation-based generalization driven by an incomplete theory of the domain--in particular by a theory for qualitative differential analysis. This paper presents an approach to constructing a planning/learning system of this kind, and is primarily organized around a hand-generated example of the type of behavior our system is intended to exhibit.

This research has been supported by NASA Ames under grant NCC 2-463, and in part by NSF under grant IRI-8740522.

Table of Contents

1. Introduction

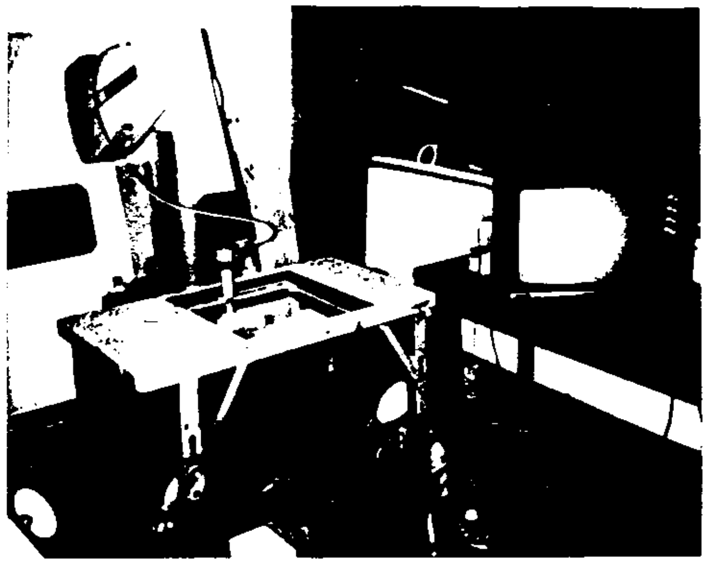
The long term goal of the work reported here is to develop robots capable of planning and acting in worlds about which they have only an incomplete theory, and capable of automatically refining their theory of the world as a result of such attempts. As a problem in machine learning, this research can be viewed as a case study in automated theory refinement. As a problem in robotics, the motivation for such research is straightforward: Current robotic systems are successful only in very carefully engineered environments in which it is feasible for the robot to know precisely the current state of its world, and in which the robot can predict accurately the results of each action that it might choose to perform. We seek more *robust* systems, which will be able to operate in less carefully engineered domains. Imagine, for example, asking a robot to "go look under that rock to see what is under it". In such a task, the robot cannot know the precise state of the world in advance (e.g., what is the size and shape of the portion of the rock which is underground?). Neither will it know with precision the consequences of actions it might perform (e.g., if it attempts to push the rock, will it budge?).

Our approach to extending the robustness of robotic systems is to allow the robot to plan relative to its approximate theory of its world and to monitor the execution of its plans. Should the robot observe that its plan is not unfolding as intended, it will seek to construct a plausible *explanation* of the possible causes of this deviation. This explanation serves as the basis for devising an error recovery strategy. It also serves as the basis for explanation-based generalization of the empirical observation, which allows refining the initial incomplete world theory.

The following sections describe in greater detail the approaches to planning with incomplete theories, plan execution monitoring, explanation-based error recovery, and automatic theory refinement. These ideas are being explored in the context of a simple hand-eye system which is described in the following section. The subsequent section describes the approach and illustrates it with a detailed, manually generated example trace. The final section summarizes the major aspects of our approach, as well as some issues raised by our initial explorations.

2. The Hand-Eye Testbed

Figure 2-1: Hand-Eye Testbed



Our research on learning robots is being conducted within a robot testbed based on a Puma 560 manipulator arm and IRI D256 vision system, as shown in figure 2-1. Both the arm and the vision system are controlled by special purpose processors that communicate with a Sun workstation running Common LISP. Common LISP functions have been developed for requesting particular arm movements, and for querying the vision system regarding the current state of the robot's world.

The problem of general manipulation is a major unsolved problem in robotics, and the task of planning, executing, monitoring, and acquiring robot manipulation strategies is a rich area for study (see, e.g., [2], [8], [4], [5], [6], [1]). The difficulty of robot manipulation stems from both the lack of observability of the world and the complexity of the computations involved. Consider, for example, the difficulty of predicting something as simple as the exact trajectory of a block lying on the table, when it is pushed by a finger from a particular point in a particular direction. The task would be burdensome but possible if one knew the exact force applied by the finger, the coefficients of friction between the finger and block and between the block and table, the precise points of contact between the block and table, the distribution of mass within the block, etc. In fact, many of these features are unobservable in general, so the task is inherently rife with uncertainty. For these reasons, we find robot manipulation an interesting domain for studying problems of planning, error recovery, and learning, under the assumption that the agent possesses only an incomplete theory of its domain.

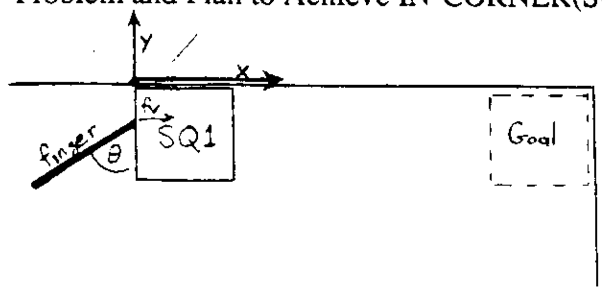
In order to avoid difficult issues of automated perception, we have chosen a very simple world for the robot. It manipulates flat objects lying on a table, by pushing these objects. The objects may collide as they are pushed, but are never picked up off the table. This essentially restricts the robot to operating in a two-dimensional world, and allows us to use simple vision algorithms to obtain reliable feedback regarding the positions, orientations, and identities of each object on the table with a delay of approximately 5 seconds. The present testbed provides a 2D world with friction (among the objects and the table), but no gravity (the objects remain in position unless pushed). We intend to alter these parameters as the project progresses, adding gravity to the 2D world by tilting the table so that objects slide to the bottom, and removing friction by utilizing a table full of holes connected to an air pump so that the objects float on a cushion of air above the table. Thus, we will be able to study a variety of worlds with and without friction or gravity, and study tasks such as constructing two-dimensional structures including arches, towers, and the like.

3. Approach

This section discusses the interrelated problems of planning, execution monitoring, error recovery, and theory refinement. Here the primary activity of the robot agent is to plan and execute actions to achieve its goals. Learning occurs in the event that planned actions fail to achieve the desired goal when they are executed. In this case, the agent attempts to explain to itself the plausible cause of the failure (in terms of a predefined theory for qualitative analysis of observed motions). This explanation is then used both for suggesting error recovery strategies and for generalizing from the observed failure to provide a more refined model of the offending action. A key assumption of the approach is that it is feasible to provide the agent a strong enough theory of its world that it can construct plausible explanations of observed failures, even though this theory may not be strong enough to predict and plan how to avoid such failures ab initio.

Figure 3-1: Problem and Plan to Achieve IN-CORNER(SQ1)

The Problem:



Where:

coordinate frame origin: x=0, y=0 is initial position of top left corner of SQ1
position of SQ1 described by <x,y,theta>, where
  x = x coordinate of top left corner of SQ1
  y = y coordinate of top left corner of SQ1
  theta = orientation relative to initial orientation of SQ1

f_x = x coordinate of tip of finger
f_y = y coordinate of tip of finger
f_{theta} = orientation of finger (independent of finger velocity)
fv = the vector velocity of the finger tip (independent of f_{theta})
$SQ1_{x\text{-}initial}$ = x coordinate of initial position of SQ1

Initial Knowledge:

To move ?object somewhere to the right
Push(?object,?f_x,?f_y,?f_{theta},?fv) with ?f_x=x coordinate of left edge of ?object

The Plan:

Initial State:
  AT(SQ1, <0,0,0>)
Goal: (target world state)
  AT(SQ1, <200,0,0>) (i.e., IN-CORNER(SQ1))
Actions: (sequence of planned actions)
  Push(SQ1, 0, -30, 60, <10, 0>)
Expectation: (predicted trajectory of world states)
  AT(SQ1, <x,y,theta>), where x > $SQ1_{x\text{-}initial}$, y<0, 0<theta<360
Intention: (target trajectory of world states)
  AT(SQ1, <x,0,0>), where x=f_x

The discussion in this section presents the approach that we are presently implementing and illustrates it in terms of a manually derived trace based on the simple planning problem shown in figure 3-1. Here the robot has the goal of moving square SQ1 into the corner, and begins with the following incomplete knowledge about the Push operation:

To move an object somewhere to the right
Push from somewhere on the left

The following subsections discuss how this knowledge can be used to produce a plausible plan for achieving the robot's goal, and how the observed failure of the plan can lead to refining this initial knowledge to the following more specialized and more precise assertion:

To *translate square SQ1 straight* to the right *along WALL2*
Push from the left *with y coordinate of fingertip at* f_y<-55

3.1. Planning

Planning is the task of generating a sequence of operations, or actions, whose expected outcome achieves the desired goal. The competence of a planner in performing this task is primarily determined by the correctness of its internal models of its actions; that is, its assumptions about each action's preconditions and postconditions. If the planner's action models are insufficiently precise, will be unable to produce a plan guaranteed to achieve the desired goal. The learning task we are considering here is the task of improving the precision and correctness of such action models.

More precisely, we view actions as mappings over states of the agent's world. An action model is generally a one-to-many mapping (e.g., the above model of the Push operation maps an initial world state into any of a set of possible outcome states). Following [5], we define a *strong plan* as a sequence of actions that maps the initial world state into a set of possible outcome states *each of which* satisfies the goal. Similarly, we define a *weak plan* as a sequence of actions that maps the initial world state into a set of outcome states *at least one of which* satisfies the goal.

While a detailed discussion of strategies for planning is beyond the scope of this paper, we intend for the agent to produce a strong plan when possible, and to produce a weak plan when strong plans are not possible. Reference [5] describes in greater detail strategies for producing such plans.

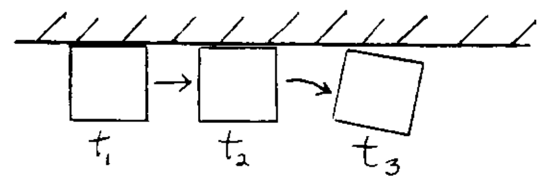
In the present example, a strong plan is not possible, since the agent's model of the Push action is insufficiently precise. Thus, the best that can be achieved is a weak plan, such as the plan shown in figure 3-1. This plan is intended to achieve the goal IN-CORNER(SQ1), but based on the agent's model of Push it can only be guaranteed to move SQ1 somewhere to the right. The plan is described in terms of the following features:

- **Goal:** the class of desired world states.
- **Actions:** a sequence of actions to be executed in order to achieve the goal.
- **Expectation:** the class of possible resulting world states, based on the agent's action models.
- **Intention:** the class of world states that are the intended effect of executing the planned actions. Note in the example plan the intentions are expressed as a function of the finger position, f_x, which varies during execution of the Push action.

3.2. Execution Monitoring

Figure 3-2: Execution Monitoring

Expectation: (predicted trajectory of world states)
    AT(SQ1, <x,y,theta>), where x > $SQ1_{x\text{-initial}}$, y<0, 0<theta<360
Intention: (target trajectory of world states)
    AT(SQ1, <x,0,0>), where x=f_x
Observations: (observed trajectory of world states)
    t_1: AT(SQ1, <0,0,0>)
    t_2: AT(SQ1, <3,0,0>)
    t_3: AT(SQ1, <6,-1,-15>) **



** Error: At $t3$, difference between intended and observed position of SQ1: <0, -1, -15>
    (i.e., intended translation (y and theta should remain 0), but observed rotation)

As the plan is executed, the vision system is used to monitor the trajectory of resulting world states. Here the vision system field of view encompasses the entire world of the robot (i.e., its table), so that we avoid questions of focus of attention in perception which appear in more general robotics domains.

What is the general condition under which plan execution should be interrupted and an error signaled? Throughout the execution of the plan, the system's observations of the world state must be consistent with the plan intention. Should this cease to be the case, the plan is no longer proceeding as intended, and an error should be signaled.

Figure 3-2 summarizes the results of executing the example plan, illustrating that the observations conflict with the plan intentions at time t_3. This corresponds to the observation that at time t_3 the square SQ1 has rotated away from the wall rather than translating directly along the wall as intended.

3.3. Error Recovery

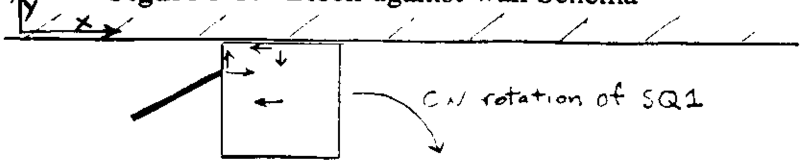
Once an error is detected in executing the plan, one might attempt to recover simply by creating a new plan to reach the goal from the newly observed world state. However, this strategy has the drawback that the factor which led to the observed error could easily foil the new plan as well (e.g., if the block rotated away from the wall because it was being pushed from the wrong pushing point, then the new plan could easily fail for the same reason). In order to avoid this difficulty, the agent first constructs a plausible explanation for the cause of the observed error, relating the cause of the error to parameters which are within its control (e.g., the parameters to the Push operator). In constructing this explanation, the agent relies on certain general knowledge of physics which forms its background domain theory. As we will see, this derived explanation can be used to suggest a revised plan that avoids the cause of the error. Furthermore, this explanation is also used as the basis for inferring the general conditions under which the error will occur, as well as general conditions under which it can be avoided. Thus, the explanation of the cause of the error is central both to recovering from the error and to learning a general refinement to the agent's action models.

3.4. Explaining the Error

Let us call the difference between the plan intention and the observations the *error feature*. We will say that the system *explains the source of the error* if it answers the question "What is the dependence of the error feature on parameters under the agent's control in earlier world states?". In the present example, the error feature is the observed but unintended rotation of square SQ1 away from the wall. Thus, in the present example the robot must explain how the rotation of SQ1 depends on the controllable parameters of its Push action.

Note that the explanation task here is fundamentally a problem of differential analysis: we are interested in the qualitative derivative of the error feature with respect to parameters under the agent's control. It is this explanation task that dictates the form of background domain theory required by the agent. If it can answer this question, then it can choose a new set of action parameters more tuned to bringing its observations in accord with its intentions. The agent derives the explanation by first identifying all possible influences on the error feature, then determining the type of dependence of the error feature on each influence, then determining which of these influences is under the agent's control.

Figure 3-3: Block-against-wall Schema



Note arrows inside SQ1 represent force components.

Applicability conditions:
IS(Block ?b)
IS(Wall ?w)
IN-CONTACT(?w ?b)

Force components acting on block	Dependence of CW rotation on force
Wall normal	? (+ or -)
Wall friction	-
Table normal	0
Table friction	-
Finger normal	+
Finger friction	+

Controllable features	Dependence of CW rotation on feature
Finger position$_y$ (i.e., $?f_y$)	+
Finger velocity$_y$ (i.e., $?fv_y$)	+
Finger velocity$_x$ (i.e., $?fv_x$)	+
Finger orientation (i.e., $?f_{theta}$)	0

Figure 3-3 illustrates the block-against-wall schema which contains the derived explanation of the observed error feature for the present example. The first part of this explanation determines all forces acting on SQ1 as well as the direction of the dependence of the error feature on each. The clockwise (CW) rotation depends positively on certain force components, negatively on others, and might depend either positively or negatively on another (the wall normal force) depending on the unobservable details of the contact between the wall and SQ1. This portion of the explanation follows from fairly straightforward knowledge of physics, based on the assumption that the force moment causes rotation, that forces arise from physical contacts, and that each physical contact gives rise to both normal and tangential force components. Since SQ1 participates in three physical contacts (with the table, the wall, and the robot finger), there are six

Figure 3-4: Primitive Qualitative Physics

- Forces arise from physical contact
- Contact forces can be characterized in terms of two components:
  - Normal component (normal to contact surface)
  - Tangential component (due to friction)
- Friction:
  - For translation, friction force directly opposes direction of translation
  - For rotation, friction resists direction of rotation
- Motion = Translation + Rotation
- Translation depends monotonically on vector sum of forces
- Rotation depends monotonically on moment of forces

force components to consider. Figure 3-4 summarizes the knowledge of physics used to construct this explanation.

Given this inferred dependence of the error feature on these force components, the second portion of the analysis determines the dependence of the error feature on features under the control of the robot (e.g., finger position, velocity, orientation). This is shown in the bottom of figure 3-3, and follows from reasoning about the geometry of the contact between the finger and SQ1 and the dependence of the resulting forces on this geometry. This final explanation of the dependence of CW rotation on the features controllable by the robot provides the key to error recovery and to theory refinement by the robot. For instance, since the explanation indicates that CW rotation is an increasing function of the y component of the finger position, one way to decrease CW rotation is to move the finger in the negative y direction. Similarly, the rotation can also be reduced by decreasing the y component of finger velocity, or by decreasing the x component of finger velocity, but not by altering the finger orientation.

The agent attempts these candidate plan revisions in order to eliminate the undesired rotation[2]. Some of these may succeed, such as moving the finger in the negative y direction. Others may

[2]In fact, the plan error may leave the world in a state such that the simple plan revision cannot be immediately applied. In this case, for example, the agent must first push SQ1 back against the wall before it can test the conjectured plan revisions.

not, such as reducing the x component of the finger velocity (which reduces the rate of rotation, but not the total rotation per unit of finger travel). When candidate plan revisions are found to succeed, then the associated explanation of the error feature receives empirical validation[3], and is used as the basis for refining the agent's action model.

3.5. Theory Refinement

Figure 3-5: Summary: Observation, Explanation and Generalization

- Observation1: CW rotation for SQ1 against WALL2 with finger at position f_y=-30, theta=60, ...
- Explanation: Sum of forces from wall, table, finger produce positive torque. CW rotation is an increasing function of finger position_y, finger velocity_y, finger velocity_x.
- Candidate Plan Revision: Move finger position_y in negative y direction to reduce rotation.
- Observation2: Pure translation (no rotation) when pushing at position f_y=-55.
- Generalization1: Pushing SQ1 against WALL2, with finger position $f_y>-30$, *any theta*, $fv_x>10, fv_y>0$ ..., will produce CW rotation.
- Generalization2: Pushing SQ1 against WALL2, with finger at position $f_y<-55$, *any theta*, $fv_x<10, fv_y<0$, ..., will produce no CW rotation.

Once the agent recovers from its error, it uses the empirically validated explanation of the cause of the error feature as the basis for explanation-based generalization of the observed failure and later success conditions. For example, as shown in figure 3-5, the explanation of the observed error feature in the present example leads to the generalization that

Pushing SQ1 against WALL2 with *finger position* $f_y>-30$, *any theta*, $fv_x>10, fv_y>0$
will produce CW rotation

This generalization is supported by the observed error (observation1 in figure 3-5), along with the (empirically supported) explanation of the dependence of the observed rotation on finger position, theta, etc. For example, since the explanation indicates that CW rotation is an increasing function of the y component of finger position f_y, and independent of the finger angle

[3]Of course the hypothesized explanation could be incorrect even if the associated error recovery strategy succeeds, since the error recovery tactic might work by coincidence or for some other reason unknown to the robot.

theta, it is reasonable to extrapolate to the generalization that rotation will occur for any value of f_y greater than the value in observation1, independent of theta. Similarly, generalization2 asserts that rotation will not occur for values of $f_y<-55$.

Note that generalization1 is only a plausible--not a guaranteed--generalization of observation1, for several reasons:

- The generalization is inferred by a form of extrapolation, inferring the behavior of SQ1 over some interval of pushing points based on only a single observation and the inferred partial derivative of rotation with respect to the pushing parameters at the current observed values of these parameters. There is in general no guarantee, for instance, that the analysis which produced the derivative of rotation with respect to f_y at f_y=-30 will hold over the entire interval $f_y>-30$, and this extrapolation is thus an approximate inference. This corresponds to an inductive bias that the derivatives of the functions of interest are slowly varying.
- The analysis of the physics is based on a number of implicit simplifying assumptions, such as the assumed uniform coefficient of friction over the surface of the table and block, the lack of jagged edges along the wall, the independence of force components and action parameters, rigid bodies, lack of inertial forces, etc. Any of these assumptions might be inappropriate, and thus the explanation on which the generalization is based might be incorrect. This corresponds to the inductive bias that the world is fairly uniform and that of all the factors influencing an object's motion only a few terms dominate.

For both of these reasons, the robot must treat its inferred generalizations as having the same status as its initial knowledge--they are plausible statements subject to subsequent empirical disconfirmation, reanalysis, and refinement. We intend for the system to retain the explanation that justifies each proposed generalization so that this explanation can be refined as needed should the generalization be empirically disconfirmed at some subsequent time. In this light, the above inductive biases lead the agent to inductive leaps that are not firm commitments, but are rather a means of delaying consideration of additional factors until some future point at which observations may indicate a more detailed analysis in warranted.

This approach of generalizing based on plausible explanations, then subsequently elaborating the analysis on a need-driven basis, is similar to the approach proposed in [9] for dealing with explanation-based learning from intractable theories in the domain of chess. In our problem domain, we anticipate that the system's ability to automatically make and later retract simplifying assumptions to ease the analysis will be an important factor in determining the overall success of the system.

4. Summary and Discussion

This paper presents our preliminary insights on the problem of developing a robot system that refines its action models and therefore improves its competence with experience. We have presented an approach to automatic refinement of robot action models based on analyzing observed plan failures in terms of a predefined qualitative theory for differential analysis. This approach is presently being implemented for a robot system based on a Puma manipulator and IRI vision system. The primary characteristics of our approach are:

- Explanation-based error recovery utilizes a plausible explanation of the source of the error to hypothesize the dependence of the error feature on controllable parameters. This hypothesized dependence suggests tactics for avoiding reoccurrences of the error while recovering from the plan failure.
- Successful error recovery lends empirical support to the hypothesized explanation, which is then used for explanation-based generalization of the observed failure and success conditions.
- Learning corresponds to demand (i.e., failure) driven refinement of an initial action model, so that the initial general-but-abstract action model is incrementally refined into a hierarchy of increasingly specialized and increasingly precise models that cover past error situations.
- Learning is guided by a theory for qualitative differential analysis. This theory is itself insufficient to entail correct plans, but is very useful for analysis of errors and guiding generalization.

4.1. Relation to Explanation Based Generalization

From the perspective of explanation based learning [3] [7], this approach utilizes a particular type of incomplete theory to produce plausible generalizations of its observations. A "classic" application of explanation based generalization [7] to this problem would require a domain theory capable of explaining/proving that the observed error feature (i.e., rotation of -15 degrees) is logically entailed by the robot's Push(SQ1, 0, -30, 60, <10,0>) action. This explanation would then be used to extract just those features of observation1 which are necessary for this prediction to hold in general. But it is unrealistic to expect that the robot could produce such an explanation here, both because the physics would be too complex, and because the explanation would depend on features such as coefficients of friction, precise contact characteristics, etc., which are *unobservable* by the vision system.

Given this difficulty in applying straightforward explanation based generalization, our approach is to instead rely on a qualitative theory for differential analysis of motions, which is less complex and does not depend on knowing precise values of physical parameters such as coefficients of friction. It is used to explain the *sign of the difference between observed and intended values of world state variables* rather than the precise observed values. This explanation is then used to suggest *directions for changing the initial action parameters* so that this error feature will be reduced. Should the suggested changes be tested and found to have the predicted effect, then the corresponding explanation receives empirical support, and is used to drive the agent's generalization process. This process extrapolates from the specific training instance to a general class of situations and action parameter values for which the action effects can be more precisely predicted. It is interesting that this theory for differential analysis is very helpful in guiding learning despite that fact that it is not useful for initial planning (e.g., it is not useful in selecting the parameters for the Push action in the initial plan). See [10] for a discussion of methods for qualitative differential analysis.

A number of questions for further research are raised by this approach: Will the generalization errors introduced by the approximate theory and the extrapolation mechanisms overwhelm the system, or will it be able to incrementally refine its beliefs as it discovers these errors during subsequent activities? Can this type of differential analysis be useful for analyzing plan execution errors that are not typically described numerically (e.g., the tower was intended to remain intact, but it fell down)? Can this differential analysis be used to automatically design feedback control loops that use the inferred dependence of the error feature on controllable parameters to continuously update the control parameters of the robot's action? Can the approach be extended to deal with situations in which the cause of the plan failure is outside the scope of the background theory (e.g., if the finger is magnetic)?

References

[1] Brost, R.C.
Automatic Grasp Planning in the Presence of Uncertainty.
In *Proceedings of the IEEE International Conference on Robotics and Automation.* IEEE , San Francisco, 1986.

[2] Carbonell, J.C., and Gil, Y.
Learning by Experimentation.
In Langley, P. (editors), *Proceedings of the Fourth International Workshop on Machine Learning*, pages 256-266. Morgan-Kaufmann, Irvine, June, 1987.

[3] DeJong, G., and Mooney, R.
Explanation-Based Learning: An Alternative View.
Machine Learning 1(2):145-176, 1986.

[4] Donald, B.R.
Error Detection and Recovery for Robot Motion Planning with Uncertainty.
Technical Report 982, MIT-AI, July, 1987.

[5] Lozano-Perez, T., Mason, M.T., Taylor, R.H.
Automatic Synthesis of Fine-Motion Strategies for Robots.
International Journal of Robotics Research 3(1):3-24, 1984.

[6] Mason, M.T.
Mechanics and Planning of Manipulator Pushing Operations.
International Journal of Robotics Research 5(1), 1986.

[7] Mitchell, T.M., Keller, R.K., and Kedar-Cabelli, S.
Explanation-Based Generalization: A Unifying View.
Machine Learning 1(1), 1986.

[8] Segre, A.M.
Explanation-Based Learning of Generalized Robot Assembly Plans.
PhD thesis, Univ. of Illinois, 1987.

[9] Tadepalli, P.
Learning Approximate Plans in Games.
December, 1986.
Rutgers Computer Science Ph.D. thesis proposal.

[10] Weld, D.
Comparative Analysis.
Artificial Intelligence :to appear, August, 1988.